

Eclipse Plug-in Development Research Project

Report 15



UNIVERSITY AT BUFFALO

November 3, 2015

Authored by: Chern Yee Chua

Eclipse Plug-in Development Research Project

Report 15

Short report

I showed the minor error messages update in email last week. I also asked some of my friends to write a simple program for me to analyze. It turns out that the work analyzing data is not as easy as I thought. It took me quite some time to analyze a small piece of code though. The reason is that the way I organize data is not optimal (even though it seems to be pretty neat). I need to switch the tables back and forth to comprehend the data that recorded in database. I am not saying it is impossible to view and rebuild the code, it is just that it takes more time than I expected. I am keeping the current one, in the meantime, I decided to redesign the plugin.

I realize that using ASTNode is a wrong choice at the beginning - there are too many restrictions using ASTNode and I am just going to name a few major issues using ASTNode:

1. Not all the codes in the editor will be accessed by traversing the root of ASTNode. For example, word such as “null”, “else”, or general comments and docs are generally not shown unless visiting the specific nodes separately.
2. While there is any syntax error, the compilation unit sometimes will be removed unintentionally.
3. It seems to recording a lot of information, but most of them are just useless. The reason why I still used the ASTNode was because I couldn't find ways to have the line number shown for each line last time.

After working some time on the new implementation, I am finally able to use line diff without relying on the line number from AST node. Now, if I add new lines of code like this:

```
5    int a;  
6    int b;  
7    int c;
```

It will return the output below:

```
+0005 int a;  
+0006 int b;  
+0007 int c;
```

The plus sign indicates line added, and minus sign is line removed. (Before that the line diff couldn't get the corresponding line number, so it needs to get the help from ASTNode, but now it can already ☺ I have tested it and I can show you how it looks like in eclipse during meeting)

It is still not enough to show the exact changes. I am thinking of doing the “git” diff style, something like this:

If int b is added in between int a and int c, I want my output to be

```
0006 int a;  
+0007 int b;  
0008 int c;
```

It will grab the line before and after the added line so that when analyzing the code, it will be very easy to view and interpret. Same for removing line, adding multiple lines, and etc. I will not use any ASTNode in this plugin but I will rely heavily on the line diff algorithm.

The final database table for compilation unit will look something like this:

date	class	Line diff	source	Error message
10/20 2.33PM	IssacClass	0006 int a; +0007 int b; 0008 int c;	Public class{ int a ; ... }	13 need “}” 14 invalid name

Note: It will still detect the error and have error messages, but it will not enter error state anymore. It will have two tables only, one is main event and the other one is compilation unit. Too many tables doesn’t really help in analyzing data.

I need few more days to finish the new design. I know what I am going to do, it is just a matter of time testing and working on the code. I believe it will be way better than the previous design if implemented successfully. If we got the permission to start recording already, we can use the old design first.

Thanks in advance!